KU LEUVEN

FACULTY OF ECONOMICS AND BUSINESS

The shooting S-estimator for robust regression

Öllerer V, Alfons A, Croux C.

KBI_1318

The shooting S-estimator for robust regression

Viktoria Öllerer [a**], Andreas Alfons [b] and Christophe Croux [a]

[a] *Faculty of Economics and Business, KU Leuven, Belgium*;

[b] *Erasmus School of Economics, Erasmus University of Rotterdam, The Nederlands*

To perform multiple regression, the least squares estimator is commonly used. However, this estimator is not robust to outliers. Therefore, robust methods such as MM-estimation have been proposed. These estimators flag any observation with a large residual as an outlier and downweight it in the further procedure. This is also the case if the large residual is caused by only one component of the observation, which results in a loss of information. Therefore, we propose the *shooting S-estimator*, a regression estimator that is especially designed for situations where a large number of observations suffer from contamination in a small number of predictor variables. The shooting S-estimator combines the ideas of the coordinate descent algorithm with simple S-regression, which makes it robust against componentwise contamination.

Keywords: cellwise outliers; componentwise contamination; shooting algorithm; coordinate descent algorithm; regression S-estimation

1. Introduction

In robust statistics it is generally assumed that the majority of observations is totally free of contamination. Any observation that deviates from the model is as a whole flagged as an outlier, even if only one component of the observation is contaminated. In case only a small number of predictor variables cause the deviation from the model, a lot of information is lost through downweighting the whole observation. Therefore, it

** Email: viktoria.oellerer@kuleuven.be

seems more appropriate to not consider whole observations as outliers but only those components that really deviate from the model. This is especially useful if the majority of observations is contaminated in only a small number of variables. Imagine, for example, a regression setting where in every observation one single predictor variable is contaminated. Here the usual robust methods break down, as there is not one single clean observation. But the majority of the cells of the design matrix is still clean and thus the majority of the data is still clean. In this setting, it is more suitable to use techniques developed for cellwise contamination (componentwise contamination) rather than those developed for rowwise contamination.

Alqallaf et al. [2009] extend the 'rowwise' contamination model to also cover cellwise contamination. They define the influence function and the breakdown point in this setting and derive them for some multivariate location estimators, showing that these cannot cope with cellwise contamination. For principal component analysis, Van Aelst et al. [2010] develop a method based on pairwise correlation that can deal with cellwise contamination. The same authors propose versions of the Stahel-Donoho estimator based on huberized outlyingness [see Van Aelst et al., 2012] and cellwise weights [see Van Aelst et al., 2011]. Affine equivariance can no longer be achieved by any of the estimation methods suitable for cellwise contamination, including ours.

In this paper we derive a regression estimator, called the *shooting S-estimator*, that can cope with cellwise contamination. It combines the ideas of the coordinate descent algorithm ('shooting algorithm') [see Friedman et al., 2007; Fu, 1998] with simple regression S-estimation [see Maronna et al., 2006]. In Section 2 we introduce the estimator and give its objective function. An algorithm is proposed in Section 3. We show simulation results in Section 4 where we compare the shooting S-estimator to the least square estimator and the robust S- and MM-estimators. Real data examples are presented in Section 5 and Section 6 concludes.

2. Objective Function

Our *shooting S-estimator* uses the idea of the coordinate descent algorithm [see Friedman et al., 2007], also called shooting algorithm [Fu, 1998]. Originally, this method performs, variable by variable, simple lasso regression. Tseng [2001] showed that by iteratively looping through all variables, it converges to the lasso estimate for any starting value. However it is well known that the lasso estimate is not robust [see e.g. Alfons et al., 2013].

In the shooting S-estimator, we achieve robustness by replacing the lasso estimation with unpenalized S-estimation [see Maronna et al., 2006]. In contrast to ordinary S-regression, the coordinate-wise approach of the coordinate descent algorithm allows us to weight all components of an observation differently.

The lasso estimate is defined as

$$\hat{\boldsymbol{\beta}}_{Lasso} = \underset{\boldsymbol{\beta}\in\mathbb{R}^p}{\arg\min}\ \frac{1}{n}\sum_{i=1}^{n}(y_i - \sum_{j=1}^{p} x_{ij}\beta_j)^2 + 2\lambda\sum_{j=1}^{p}|\beta_j|.$$

In the coordinate descent algorithm, to compute the lasso coefficient $\hat{\beta}_j$ $(j = 1, \ldots, p)$, all other coefficients are kept fixed at $\tilde{\beta}_k$ $(k \neq j)$

$$\begin{aligned}\hat{\beta}_{j,Lasso} &= \underset{\beta_j\in\mathbb{R}}{\arg\min}\ \frac{1}{n}\sum_{i=1}^{n}((y_i - \sum_{k\neq j} x_{ik}\tilde{\beta}_k) - x_{ij}\beta_j)^2 + 2\lambda\sum_{k\neq j}|\tilde{\beta}_k| + 2\lambda|\beta_j| \\ &= \underset{\beta_j\in\mathbb{R}}{\operatorname{argmin}}\ \frac{1}{n}\sum_{i=1}^{n}((y_i - \sum_{k\neq j} x_{ik}\tilde{\beta}_k) - x_{ij}\beta_j)^2 + 2\lambda|\beta_j|. \qquad (1)\end{aligned}$$

This can be seen as simple lasso regression where the response

$$\tilde{y}_i^{(j),cd} = y_i - \sum_{k\neq j} x_{ik}\tilde{\beta}_k, \quad i = 1, \ldots, n, \qquad (2)$$

is regressed on x_{ij}, for a fixed value of j.

For our shooting S-estimator, we want to make sure that the new response $\tilde{y}_i^{(j)}$ is not influenced by outliers in the cells x_{ik}. Therefore, we add cell weights w_{ik} to (2):

$$\tilde{y}_i^{(j)} = y_i - \sum_{k\neq j} w_{ik}x_{ik}\tilde{\beta}_k, \quad i = 1, \ldots, n. \qquad (3)$$

These cell weights we define as

$$w_{ik} := w(\frac{\tilde{y}_i^{(k)} - x_{ik}\hat{\beta}_k}{\hat{\sigma}_k}) \qquad (4)$$

where the argument of the weighting function $w(\cdot)$ is the scaled residual of regressing $\tilde{y}_i^{(k)}$ on x_{ik}. We choose the weighting function w to correspond to the loss function ρ to be used in the simple S-estimation step, so

$$w(z) = \frac{\frac{\partial}{\partial z}\rho(z)}{z}, \qquad (5)$$

but other choices are possible as well. It is important to note that the cell weights are defined implicitly and depend on the regression estimate.

To finally compute the estimate $\hat{\beta}_j$, we use instead of the lasso as in (1), the robust unpenalized simple S-regression. So

$$\hat{\beta}_j = \arg\min_{\beta \in \mathbb{R}} \hat{\sigma}_j(\beta)$$
$$\text{with } \frac{1}{n}\sum_{i=1}^{n} \rho(\frac{\tilde{y}_i^{(j)} - x_{ij}\beta_j}{\hat{\sigma}_j(\beta)}) = \delta.$$

Here δ equals the expected value of the ρ-function at the normal distribution, i.e. $\delta = \mathbb{E}[\rho(Z)]$ with $Z \sim \mathcal{N}(0,1)$. It is chosen so that the breakdown point (BP) of our estimator is not too low, while its efficiency is high enough. As a ρ-function we will use either Tukey's biweight

$$\rho_{BI}(z) = \begin{cases} \frac{k_{BI}^2}{6}(1-(1-(\frac{z}{k_{BI}})^2)^3) & \text{if } |z| \le k_{BI} \\ \frac{k_{BI}^2}{6} & \text{if } |z| > k_{BI}, \end{cases} \tag{6}$$

or the skipped Huber

$$\rho_{skH}(z) = \begin{cases} \frac{1}{2}z^2 & \text{if } |z| \le k_{skH} \\ \frac{k_{skH}^2}{2} & \text{if } |z| > k_{skH}. \end{cases} \tag{7}$$

These two ρ-functions are quite different in nature. The skipped Huber loss is quadratic in a central region $[-k_{skH}, k_{skH}]$ and constant outside this interval. Thus, skipped Huber is a skipped version of the quadratic loss. Its weight function (5) takes only values from $\{0,1\}$. In contrast, the biweight loss is designed to be smooth while still bounding the effect of extreme values. Its weight function (5) can take any value in $[0,1]$. Apart from those two loss functions any ρ-function [see Maronna et al., 2006, p31, Def 2.1] could be used as well.

This leads us to the objective function of our *shooting S-estimator*

$$\hat{\boldsymbol{\beta}} = \arg\min_{\beta \in \mathbb{R}^p} \sum_{j=1}^{p} \hat{\sigma}_j(\boldsymbol{\beta}) \tag{8}$$
$$\text{with } \frac{1}{n}\sum_{i=1}^{n} \rho(\frac{y_i - \sum_{k\neq j} w_{ik}(\boldsymbol{\beta}) x_{ik}\beta_k - x_{ij}\beta_j}{\hat{\sigma}_j(\boldsymbol{\beta})}) = \delta$$
$$\text{and } w_{ik}(\boldsymbol{\beta}) = w(\frac{y_i - \sum_{\ell\neq k} w_{i\ell}(\boldsymbol{\beta}) x_{i\ell}\beta_\ell - x_{ik}\beta_k}{\hat{\sigma}_k(\boldsymbol{\beta})}).$$

3. Algorithm

To compute the shooting S-estimate described in Section 2, i.e. the minimizer of Equation (8), we use an iterative procedure similar to the coordinate descent algorithm. One coefficient $\hat{\beta}_j$ is updated at a time while all other coefficients $\hat{\beta}_k$ $(k \neq j)$ are kept fixed. We do this looping through all variables $j = 1, \ldots, p$ repeatedly until convergence of the coefficient $\hat{\boldsymbol{\beta}} = (\hat{\beta}_1, \ldots, \hat{\beta}_p)$. To compute the update of the coefficient $\hat{\beta}_j$, we perform simple S-regression of $\tilde{y}_i^{(j)}$ on x_{ij}.

Algorithm 1 gives a detailed description of the algorithm used. In each step of the coordinate descent loop, we compute the simple regression S-estimate following the iteratively reweighted least squares (IRLS) algorithm recommended by Maronna et al. [2006]. Thereby, we follow the ideas of the fast-S algorithm [see Salibian-Barrera and Yohai, 2006]. We start from an initial value in the very first step and from the estimates of the previous coordinate descent loop in any other step. Then we compute an I-step which is one step of IRLS. The parameter estimate $\hat{\beta}_j$ can be directly computed via weighted least squares. Afterwards we compute the corresponding M-scale through fixed-point iteration $f(s) := 1/(n\delta) \sum_{i=1}^{n} \rho((\tilde{y}_i^{(j)} - x_{ij}\hat{\beta}_j)/s)s = s$. As starting value s_0 of the fixed-point iteration, we use the Median Absolute Deviation (MAD) of the residuals. This is repeated until convergence of the parameter estimate $\hat{\beta}_j$. Using Lemma 1 of Salibian-Barrera and Yohai [2006], we can show that the objective function (8) decreases in each step. Afterwards we update the cell weights w_{ij} with $i = 1, \ldots, n$ as in (4). Then the new response $\tilde{y}_i^{(j)}$ is computed as in (3). We perform this coordinate descent loop repeatedly for all variables $j = 1, \ldots, p$ until convergence.

Ordering the variables differently in the coordinate descent loop does not have any significant effect on the value of the shooting S-estimate. For simplicity, we therefore stick with the order $j = 1, \ldots, p$. Better convergence may be reached by ordering the variables according to the size of the initial parameter estimate $\hat{\boldsymbol{\beta}}^{(0)}$.

Before we can start the algorithm, we need to initialize the parameter estimates. We choose the extremely robust median of slopes

$$\hat{\beta}_j^{(0)} = \text{median}(\frac{y_1}{x_{1j}}, \ldots, \frac{y_n}{x_{nj}}) \qquad j = 1, \ldots, p.$$

To get the starting weights, we apply the weight function (5) to the robustly standardized x_{ij}- and y_i-values $(i = 1, \ldots, n; j = 1, \ldots, p)$. Each cell weight w_{ij} we then define as the

minimum of these two values

$$w_{ij} = \min(w(x_{ij}), w(y_i)).$$

The main difference to other robust estimates like multiple regression S-estimates is the possibility to use different weights for the components of an observation. This makes the shooting S-estimate applicable for a cellwise contamination setting.

Algorithm 1. ***Computation of shooting S-estimate***

 # Standardization

- $\mu_j^X := \text{median}(x_{1j}, \ldots, x_{nj}), \quad j = 1, \ldots, p$
- $\sigma_j^X := \text{MAD}(x_{1j}, \ldots, x_{nj}), \quad j = 1, \ldots, p$
- $x_{ij} := \frac{x_{ij} - \mu_j^X}{\sigma_j^X}, \quad i = 1, \ldots, n,\ j = 1, \ldots, p$
- $\mu^Y := \text{median}(y_1, \ldots, y_n)$
- $y_i := y_i - \mu^Y, \quad i = 1, \ldots, n$

 # Initialization

- $L := 0$ *# Number of steps in coordinate descent loop*
- $\hat{\beta}_j^{(0)} := \text{median}(\frac{y_1}{x_{1j}}, \ldots, \frac{y_n}{x_{nj}}), \quad j = 1, \ldots, p$
- $w_{ij}^X = w(x_{ij}), \quad i = 1, \ldots, n,\ j = 1, \ldots, p$
- $w_i^Y = w(\frac{y_i}{\text{MAD}(y_1, \ldots, y_n)}), \quad i = 1, \ldots, n$
- $w_{ij}^{(0)} = \min(w_{ij}^X, w_i^Y), \quad i = 1, \ldots, n,\ j = 1, \ldots, p$
- $\tilde{y}_i := y_i - \sum_{j=1}^p w_{ij}^{(0)} x_{ij} \hat{\beta}_j^{(0)}$

 # Coordinate descent loop

- $L := L + 1$
- For $j = 1, \ldots, p$ *# Index of the variable used in regression step*

 # Regression step

- $\tilde{y}_i := \tilde{y}_i + w_{ij}^{(L-1)} x_{ij} \hat{\beta}_j^{(L-1)}, \quad i = 1, \ldots, n$
- $r := 0$ *# Number of I-steps*
- $w_{ij}^{(L,0)} := w_{ij}^{(L-1)}$

 # I-step

- $r := r + 1$
- $\hat{\beta}_j^{(L,r)} := \arg\min_{\beta \in \mathbb{R}} \sum_{i=1}^n w_{ij}^{(L,r-1)} (\tilde{y}_i - x_{ij}\beta)^2$
  # Solved by weighted least squares
- $res_i^{(L,r)} := \tilde{y}_i - x_{ij} \hat{\beta}_j^{(L,r)}, \quad i = 1, \ldots, n$
- $\ell := 0$ *# Number of M-steps*
- $s_0 = \begin{cases} \text{median}(|res_1^{(L,r)}|, \ldots, |res_n^{(L,r)}|)/0.6745 & \text{if } r = 1 \\ s_j^{(L,r-1)} & \text{if } r > 1 \end{cases}$

 # M-step

- $\ell := \ell + 1$
- $s_\ell := \sqrt{\frac{s_{\ell-1}^2}{\delta \cdot n} \sum_{i=1}^n \rho(\frac{res_i^{(L,r)}}{s_{\ell-1}})}$

- Repeat M-step until $|\frac{s_\ell}{s_{\ell-1}} - 1| < \epsilon$

- $s^{(L,r)} := s_\ell$
- $w_{ij}^{(L,r)} := w(\frac{res_i^{(L,r)}}{s^{(L,r)}}), \quad i = 1, \ldots, n$

- Repeat I-step until $\max_{i=1,\ldots,n} |res_i^{(L,r)} - res_i^{(L,r-1)}| < \epsilon$
  $\rightsquigarrow w_{ij}^{(L)} := w_{ij}^{(L,r)} \quad i = 1, \ldots, n$
  $\rightsquigarrow \hat{\beta}_j^{(L)} := \hat{\beta}_j^{(L,r)}$
- $\tilde{y}_i := \tilde{y}_i - w_{ij}^{(L)} x_{ij} \hat{\beta}_j^{(L)}, \quad i = 1, \ldots, n$

- $resid_i^{(L)} := \tilde{y}_i, \quad i = 1, \ldots, n$
- Repeat coordinate descent loop until $\max_{i=1,\ldots,n} |resid_i^{(L)} - resid_i^{(L-1)}| < \epsilon$
  $\rightsquigarrow \hat{\beta}_j^{centered} := \hat{\beta}_j^{(L)}$

 # Backtransformations
- $\hat{\beta}_j := \hat{\beta}_j^{centered} / \sigma_j^X, \quad j = 1, \ldots, p$
- $\hat{\beta}_0 := \mu^Y - \sum_{j=1}^p \hat{\beta}_j \mu_j^X$
- $\hat{\boldsymbol{\beta}} := (\hat{\beta}_0, \hat{\beta}_1, \ldots, \hat{\beta}_p)$

4. Simulation Setup

To evaluate the shooting S-estimator, we compare it to the classical least squares estimator (LS), the ordinary S-estimator and the MM-estimator [see Maronna et al., 2006]. For the shooting S-estimator we use once the biweight ρ-function (6) and once the skipped Huber ρ-function (7). For the parameter k in the ρ-function, we choose $k_{BI} = 3.420$ and $k_{skH} = 2.177$. This corresponds to a breakdown point of 20% in the simple regressions. Our choice seems to be a good trade-off between robustness and efficiency. For the computation of the ordinary S-estimate, we use the biweight loss function and set again $k_{BI} = 3.420$. The MM-estimate is computed with the standard settings of 50% breakdown point and an efficiency of 95% at the normal model, using the biweight loss function. We stick here to the high breakdown point of 50%, as MM can achieve high efficiency and a high breakdown point simultaneously. Thus, lowering the breakdown point would not increase the efficiency of the MM-estimator.

For the simulation setup we set $n = 100$ and $p = 15$. The regression coefficients $\boldsymbol{\beta}$ are taken equally spaced over the interval [0,1], i.e. $\beta_j = j/p$ for $j = 1, \ldots, p$. The predictors $\mathbf{x}_i$ and errors e_i are independent and identically normally distributed with mean 0 for $i = 1, \ldots, n$. We choose two different sampling schemes, one with uncorrelated and one with correlated predictors. For the first one, we use the identity matrix as a covariance matrix for the predictors. The error variance is $\sigma^2 = 0.5^2$. In the correlated setting

we choose the predictor covariance matrix Σ with $\Sigma_{ij} = 0.5^{|i-j|}$ and the error variance $\sigma^2 = 0.81^2$. By this the signal-to-noise ratio[1] is the same in both settings. The response variable is then created as $y_i = \mathbf{x}_i'\boldsymbol{\beta} + e_i$ for $i = 1, \ldots, n$.

To every generated data set we add 1%, 2%, 5% and 10% of cellwise contamination. The cells x_{ij} that we contaminate are chosen randomly from the design matrix X. So every cell of our data set is equally likely to be contaminated. Three different contamination settings are used: a dense cluster $x_{ij}^{cont} \sim \mathcal{N}(50, 1)$, scattered outliers $x_{ij}^{cont} \sim \mathcal{N}(0, 100^2)$ and a wide cluster $x_{ij}^{cont} \sim \mathcal{N}(50, 10^2)$. We only contaminate the x-values and not the y-values, which creates bad leverage points. For comparison, we also construct classical contamination settings where we choose whole rows for contamination instead of cells. For these we choose the three contaminations $\mathbf{x}_i^{cont} \sim \mathcal{N}(\mathbf{50}, \Sigma)$, $\mathbf{x}_i^{cont} \sim \mathcal{N}(\mathbf{0}, 100^2 \cdot \Sigma)$ and $\mathbf{x}_i^{cont} \sim \mathcal{N}(\mathbf{50}, 10^2 \cdot \Sigma)$. Additionally, we also want to demonstrate that the shooting S-algorithm can deal with contamination in the response. From the clean data set, we select 1%, 2%, 5% and 10% of observations and contaminate their error terms $e_{cont} \sim \mathcal{N}(50, \sigma^2)$ to create vertical outliers.

To compare the different estimators, we apply them to $R = 1000$ generated data sets. For each data set we compute the mean squared error (MSE)

$$\text{MSE}(\hat{\boldsymbol{\beta}}) = \frac{1}{p}\sum_{j=1}^{p}\frac{1}{R}\sum_{r=1}^{R}(\hat{\beta}_j^{(r)} - \beta_j)^2.$$

Additionally, also the bias or the median squared error could be used as evaluation methods. We omit them as they are in line with the MSE.

The simulation results for cellwise contamination are displayed in Tables 1 and 2 for uncorrelated and correlated predictors, respectively. Table 3 gives the results for rowwise contamination in the data set with correlated predictors. Table 4 illustrates the behavior of the estimators in presence of vertical outliers for correlated predictors. The standard errors around the reported results are smaller than 3% in all tables. We omit the results for rowwise contamination and vertical outliers for uncorrelated predictors as they are comparable to the ones in the correlated case.

For uncorrelated predictors, Table 1 demonstrates the need of a new method that can deal with cellwise contamination. As well known, LS breaks down for any amount of contamination. But also the robust MM- and S-estimator have problems with larger amounts of cellwise contamination. As 2% of cellwise contamination correspond here

[1]The signal-to-noise ratio equals $\frac{\sqrt{\boldsymbol{\beta}'\Sigma\boldsymbol{\beta}}}{\sigma}$.

Table 1: $n \cdot MSE$ of different estimators for cellwise contamination for all three contamination settings with $n = 100$, $p = 15$ and uncorrelated predictors

		$\epsilon = 0$	$\epsilon = 0.01$	$\epsilon = 0.02$	$\epsilon = 0.05$	$\epsilon = 0.1$
$x_{ij}^{cont} \sim \mathcal{N}(50, 1)$	LS	0.30	23.75	31.66	35.95	36.47
	S	0.36	1.11	12.82	32.30	36.36
	MM	0.33	0.49	0.91	18.12	34.42
	shooting S + BI	1.04	1.41	1.68	2.52	5.02
	shooting S + skH	0.47	0.86	1.14	2.04	4.20
$x_{ij}^{cont} \sim \mathcal{N}(0, 100^2)$	LS	0.31	22.21	30.61	36.11	36.66
	S	0.36	1.00	11.11	31.09	36.54
	MM	0.33	0.49	0.99	17.12	33.37
	shooting S + BI	1.05	1.52	1.90	4.21	12.90
	shooting S + skH	0.48	0.86	1.21	2.71	7.13
$x_{ij}^{cont} \sim \mathcal{N}(50, 10^2)$	LS	0.30	23.81	31.44	35.95	36.47
	S	0.36	1.08	12.89	32.51	36.43
	MM	0.33	0.49	0.93	18.53	34.13
	shooting S + BI	1.04	1.47	1.73	2.81	6.44
	shooting S + skH	0.46	0.88	1.15	2.11	4.90

to about $20 - 30\%$ of rowwise contamination[2], the ordinary S-estimator already breaks down. As we have chosen a breakdown point of 50% for MM, it can deal with slightly higher contamination. But for about $4 - 5\%$ of cellwise contamination it also breaks down, as this means that more than 50% of the observations are contaminated. In contrast, the shooting S-estimators can deal with much higher levels of cellwise contamination. They are reliable even for up to 10% of cellwise contamination, which corresponds to up to 80% of rowwise contamination. The two shooting S-estimators perform comparably in this setting. Admittedly, the shooting S-estimators loose precision for low levels of contamination compared to the other estimators. For the different contamination settings chosen, the shooting S-estimators perform best for the dense cluster and worst for scattered outliers.

Table 2 confirms for correlated predictors what is shown in Table 1 for uncorrelated

[2]The expected value of the number of contaminated rows is $n(1-(1-\epsilon)^p)$ for a cellwise contamination level ϵ.

Table 2: $n \cdot MSE$ of different estimators for cellwise contamination for all three contamination settings with $n = 100$, $p = 15$ and correlated predictors

		$\epsilon = 0$	$\epsilon = 0.01$	$\epsilon = 0.02$	$\epsilon = 0.05$	$\epsilon = 0.1$
$x_{ij}^{cont} \sim \mathcal{N}(50, 1)$	LS	1.29	35.59	39.85	35.55	36.01
	S	1.55	6.27	21.48	45.41	40.50
	MM	1.40	2.77	5.65	27.04	46.61
	shooting S + BI	1.78	4.01	4.94	5.85	7.62
	shooting S + skH	1.79	8.33	10.95	12.28	12.70
$x_{ij}^{cont} \sim \mathcal{N}(0, 100^2)$	LS	1.31	32.56	38.72	36.14	36.48
	S	1.57	5.55	19.28	42.92	40.45
	MM	1.43	2.87	5.52	25.06	43.72
	shooting S + BI	1.81	3.96	5.48	9.13	17.24
	shooting S + skH	1.82	7.05	10.26	14.48	18.44
$x_{ij}^{cont} \sim \mathcal{N}(50, 10^2)$	LS	1.29	35.42	39.35	35.41	36.02
	S	1.55	6.37	21.24	45.12	40.79
	MM	1.41	2.92	5.56	27.15	46.52
	shooting S + BI	1.79	4.12	5.40	6.48	9.17
	shooting S + skH	1.81	8.42	11.53	12.86	13.85

ones. The only major difference is that for correlated predictors the shooting S-estimator using a biweight loss is performing best. It can also be noted that in this setting, for both versions of clustered outliers, the shooting S-estimator outperforms the MM-estimator already for 2% of cellwise contamination, even though MM does not break down yet in this case.

For rowwise contamination the situation is different (see Table 3). Here MM and S-estimation give excellent results. Shooting S-estimation gives similar results as for the cellwise contamination setting. This means that for rowwise contamination, shooting S-estimation cannot compete with ordinary S- or MM-estimation. Therefore, we do not advise to use the shooting S-estimator if only rowwise contamination is present.

The shooting S-estimator can also cope with vertical outliers (see Table 4). It gives good results for all levels of contamination used here, although its MSE is slightly higher than for the S- and MM-estimators. The reason for the good performance of the shooting S-estimator is that the contamination in the response is present in the computation of

Table 3: $n \cdot MSE$ of different estimators for rowwise contamination for all three contamination settings with $n = 100$, $p = 15$ and correlated predictors

		$\epsilon = 0$	$\epsilon = 0.01$	$\epsilon = 0.02$	$\epsilon = 0.05$	$\epsilon = 0.1$
$x_{ij}^{cont} \sim \mathcal{N}(50, 1)$	LS	1.30	53.37	54.36	54.08	54.12
	S	1.56	1.54	1.48	1.47	1.48
	MM	1.42	1.43	1.39	1.43	1.50
	shooting S + BI	1.81	4.92	5.92	7.00	9.49
	shooting S + skH	1.80	11.64	13.96	15.16	15.48
$x_{ij}^{cont} \sim \mathcal{N}(0, 100^2)$	LS	1.26	12.80	22.03	44.31	57.15
	S	1.51	1.57	1.63	1.74	2.27
	MM	1.38	1.45	1.53	1.55	1.81
	shooting S + BI	1.77	2.83	4.20	8.19	13.55
	shooting S + skH	1.78	4.32	7.32	14.09	18.06
$x_{ij}^{cont} \sim \mathcal{N}(50, 10^2)$	LS	1.31	57.22	55.85	54.56	49.53
	S	1.55	1.49	1.51	1.45	1.50
	MM	1.42	1.38	1.42	1.41	1.51
	shooting S + BI	1.82	4.90	5.67	7.27	11.26
	shooting S + skH	1.81	11.52	13.42	15.67	17.55

each single coefficient $\hat{\beta}_j$, which makes it easy for the method to detect the outlier.

We may conclude that the shooting S-estimator is the only regression estimator that can deal with cellwise contamination above 5%. The estimator also gives good results in presence of vertical outliers. However, in a rowwise contamination setting, the usual robust methods as S- and MM-estimation are to prefer.

5. Real Data

We also evaluate the performance of the shooting S-estimator on real data sets. To this end, we choose the three data sets `Cars93`, `Auto` and `Boston`. The `Cars93` data, a selection of 1993 model cars, are included in the R package `MASS`. Omitting not fully observed data points, we are left with $n = 82$ observations. Using the same estimators as in Section 4, we fit the following model including $p = 22$ variables of the `Cars93` data

Table 4: $n \cdot MSE$ of different estimators for vertical outliers with $n = 100$, $p = 15$ and correlated predictors

	$\epsilon = 0$	$\epsilon = 0.01$	$\epsilon = 0.02$	$\epsilon = 0.05$	$\epsilon = 0.1$
LS	1.28	49.67	99.88	233.83	445.79
LTS	3.32	3.30	3.24	3.12	2.72
S	1.51	1.52	1.49	1.47	1.46
MM	1.38	1.40	1.40	1.44	1.47
shooting S + BI	1.75	1.82	1.83	1.93	2.45
shooting S + skH	1.79	1.83	1.81	1.93	2.40

(for an overview see Table 7 in the appendix)

$$\begin{aligned} PRICE =& \beta_0 + \beta_1 MAN + \beta_2 TYPE + \beta_3 MPG.C + \beta_4 MPG.H \\ &+ \beta_5 AIRBAGS + \beta_6 DT + \beta_7 CYLIN + \beta_8 ENG.SIZE \\ &+ \beta_9 HP + \beta_{10} RPM + \beta_{11} REV.MILE + \beta_{12} MAN.TRANS \\ &+ \beta_{13} FUEL.TANK + \beta_{14} PASSENGER + \beta_{15} LENGTH \\ &+ \beta_{16} WHEELBASE + \beta_{17} WIDTH + \beta_{18} TURN \\ &+ \beta_{19} REAR.SEAT + \beta_{20} LUGGAGE + \beta_{21} WEIGHT \\ &+ \beta_{22} ORIGIN + error. \end{aligned}$$

When applying the shooting S-estimator, we declare a component of an observation, hence a cell in the data matrix, as an outlier if it gets a weight w_{ij} below 0.5. If all components of an observation are flagged as outliers, we say that the whole observation is outlying. The shooting S-estimator with a skipped Huber loss characterizes two observations as a whole as outliers, while it flags for 14 observations only a few cells as outliers. In contrast, the shooting S-estimator with biweight loss detects only for one observation a single outlying cell and downweights 12 observations as a whole. Both shooting S-estimators declare more observations as a whole as an outlier than the ordinary S- and MM-estimator, who give only three observations a weight below 0.5. Those three observations are also downweighted by the shooting S-estimators.

The `Auto` data set is included in Stata and can be downloaded from `http://www.stata-press.com/data/r13/auto.dta` [see StataCorp, 2013]. It consists of $n = 69$ fully observed sales of vintage 1978 automobiles in the United States (see Table 8 in the

appendix). We fit the following model consisting of $p = 10$ variables

$$\begin{aligned} PRICE =& \beta_0 + \beta_1 MPG + \beta_2 REP78 + \beta_3 HEADROOM + \beta_4 TRUNK \\ &+ \beta_5 WEIGHT + \beta_6 LENGTH + \beta_7 TURN + \beta_8 DISPLACE \\ &+ \beta_9 GEAR + \beta_{10} FOREIGN + error. \end{aligned}$$

This data set was also used by Verardi and Croux [2009], who discovered vertical outliers as well as bad and good leverage points. Both shooting S-estimators find all of these bad leverage points and vertical outliers, while giving high weights to the good leverage points. In addition, both shooting S-estimators (as well as the ordinary S- and MM-estimator) also flag observation 'Buick Riviera' as an outlier. Furthermore, the shooting S-estimator with biweight loss gives for some observations (Cadillac Deville, Lincoln Continental, Olds Toronado, Audi 5000, BMW 320i) considerably lower weight to one component, but not the others, indicating different behavior of single cells. The MM-estimator cannot distinguish between the different components of an observation, thus, it downweights all those observations as a whole. In contrast, the ordinary S-estimator does not identify those observations as outlying. Also the shooting S-estimator with skipped Huber loss does not downweight them. A reason may be that the skipped Huber weighting function can only give values in $\{0, 1\}$, which can cause problems if observations only moderately deviate from the majority of the data.

The third data set, the `Boston` housing data, originates from Harrison and Rubinfeld [1978] and has been extensively analyzed in the robust statistics literature. The data is available in the R package `mlbench` and contains various characteristics of houses, demographics, air pollution and geographical details on $n = 506$ census tracts in and nearby Boston. Table 9 in the appendix gives an overview of the $p = 9$ variables used to fit the model

$$\begin{aligned} \log(MEDV) =& \beta_0 + \beta_1 CRIM + \beta_2 NOX^2 + \beta_3 RM^2 + \beta_4 AGE + \beta_5 \log(DIS) \\ &+ \beta_6 TAX + \beta_7 PTRATIO + \beta_8 B + \beta_9 \log(LSTAT) + error. \end{aligned}$$

Belsley et al. [1980] discovered outlying behavior of census tracts lying in central area of Boston, and show that the influential census tracts are heavily concentrated in a few neighborhoods. Applying the shooting S-estimators to the full data set, we get similar results: Most of the outlying observations lie within the center of Boston, and and census tracts of the same neighborhood are usually jointly indicated as outliers. Both

shooting S-estimators declare only some components of the observations $365 - 370$ (all from the neighborhood Back Bay) as contaminated, indicating outlyingness of single components and not of whole observations. But as the number of observations in this data set is sufficiently large ($n = 506$ while $p = 9$), using the clean components of those 6 observations in the estimation results only in an small gain of information. Thus, the shooting S-estimators and the MM-estimator perform comparably here.

To evaluate the performance of the shooting S-method, we compute the shooting S-estimate on each of the three data sets, once combined with biweight loss and once with skipped Huber loss, and compare them to the LS, S- and MM-estimates. (For all estimators, all parameters are chosen as in Section 4.) For each of the three data sets, we randomly choose 4/5th of the observations and compute all estimates on this training data set. This we repeat $R = 500$ times and compare the estimates on the training data sets $\hat{\boldsymbol{\beta}}^{(r)}$ to the ones computed on the full data set $\hat{\boldsymbol{\beta}}^{full}$ through the Average Norm Distance (AND), which we adjust for the different scales of the explanatory variables $\mathbf{x}_{\cdot j}$

$$AND(\hat{\boldsymbol{\beta}}) = \frac{1}{R}\sum_{r=1}^{R}\sqrt{\sum_{j=1}^{p}(\hat{\beta}_j^{(r)} - \hat{\beta}_j^{full})^2\,\mathrm{MAD}(\mathbf{x}_{\cdot j})^2}. \tag{9}$$

As an estimator is considered robust if it is not too much influenced by single observations, a low value of AND is desired. Table 5 shows the results for all five estimators on all three data sets. It can be seen that the shooting S-estimator combined with a biweight loss gives lowest results for all three data sets. Surprisingly, the shooting S-estimator using a skipped Huber loss performs worst. It chooses weights only from $\{0, 1\}$ which can cause instability in real data applications. A skipped Huber loss may also not be advisable for correlated predictor variables (as we could see from the simulation study in Section 4).

Table 5: Average norm distance (AND) of estimates on the training data set and the full data set for all three real data examples

	LS	S	MM	shooting S + BI	shooting S + skH
Auto (n = 69, p = 10)	1494.463	1339.269	1588.926	1130.745	1723.673
Cars93 (n = 82, p = 22)	5.434	5.451	6.001	2.944	8.498
Boston (n = 506, p = 9)	0.022	0.019	0.020	0.016	0.032

A robust estimator should not change considerably if small amounts of contamination

are added to the data. Therefore, we investigate the change of the various estimates under additional cellwise contamination. To this end, we randomly choose 5% of the cells and replace them with $x_{ij}^{cont} \sim \mathcal{N}(\hat{\mu}_j + 10\hat{\sigma}_j, \hat{\sigma}_j^2)$ where $\hat{\mu}_j$ and $\hat{\sigma}_j$ denote the median and MAD of the jth column of the design matrix X, respectively. This we repeat $R = 500$ times and compute the average norm difference (9) of the estimates on the contaminated data $\hat{\boldsymbol{\beta}}^{(r)}$ and the estimate on the original data $\hat{\boldsymbol{\beta}}^{full}$. Table 6 gives the results. For the `Auto` and the `Boston` data, the MM-estimator gives lowest values, while the shooting S-estimator with biweight loss performs similarly as the ordinary S-estimator and shooting S with skipped Huber loss slightly worse. For the `Cars93` data, shooting S-biweight gives lowest results of all estimators. One should be aware that, to be fair, the shooting S-estimator should only be compared to the S-estimator, as only those two are similar in nature (e.g. 20% breakdown point for shooting S, while MM has 50%). As we see form Table 6, the shooting S estimator with biweight loss yields similar values of the AND as the corresponding ordinary S-estimator for `Auto` and `Boston` and outperforms it for `Cars93`. A shooting MM-estimator may show better performance than the ordinary MM-estimator. But this subject is left for further research.

Table 6: Average norm distance (AND) of the estimates on the contaminated data sets and the clean data set for all three real data examples

	LS	S	MM	shooting S + BI	shooting S + skH
Auto (n = 69, p = 10)	5025.669	2424.730	1256.571	2425.611	2516.561
Cars93 (n = 82, p = 22)	13.515	8.825	8.796	6.062	9.418
Boston (n = 506, p = 9)	0.253	0.207	0.168	0.208	0.210

6. Conclusion

In this paper, we introduce a regression estimator applicable for cellwise contamination. It combines the ideas of ordinary regression S-estimation with the coordinate descent algorithm. Thereby the shooting S-estimator is able to use different weights for different components of an observation. In our simulations, it can deal with cellwise contamination up to 10%.

Furthermore, the shooting S-estimator can also be used as a diagnostic tool. After computation of the shooting S-estimate, the entries of the weight matrix w_{ij} help to

distinguish between cellwise and rowwise contamination. If all components of the same observations get similar weights, this indicates that the whole observation is contaminated or that it is a vertical outlier. Thus, no cellwise contamination is present and therefore we do not advise the shooting S-estimator to perform regression analysis. Instead a robust method such as MM-estimation should be used.

The shooting S-estimator suffers from a lack of efficiency, especially in the setting of rowwise contamination. Therefore it might be interesting for further research to develop a *shooting MM-estimator*. This means to replace in the shooting S-estimator the step of simple S-estimation with simple MM-estimation.

Another idea worth considering is the application of an imputation method after performing shooting S-regression. Cells that are flagged as outliers can be set as missing. On the data set containing missing values, regression can be performed [see Little, 1992].

Admittedly, our shooting S-estimator has problems with good leverage points. Particularly, if not a lot of contamination is present, the shooting S-estimator tends to flag good leverage points as outliers. Impreciseneses in the starting value of the estimate can propagate throughout the algorithm leading to a large residual. This is especially likely if some of the observed components are large in absolute value, which is usually the case for good leverage points. However, in the real data examples of Section 5 this did not happen.

As the shooting S-algorithm deals with each variable separately, it can also be applied to data sets with a small sample size and even if $n < p$. A suitable ρ-function for this setting may be the linear quadratic quadratic (lqq) function of Koller and Stahel [2011], as it has been shown to have high efficiency also for small sample sizes. When using the lqq-function in the simulations setups in Section 4, the results are comparable to the ones with the other ρ-functions used there.

Finally, the shooting S-estimator can be extended to a *penalized shooting S-estimator*. To the simple S-estimation in every variable, a penalty term $J(\beta)$ can be added. Possible choices for the penalty term are $J(\beta) = |\beta|$ or $J(\beta) = \beta^2$. The penalized version of the shooting S-estimator could be very useful in high-dimensional settings.

A. APPENDIX - Description of Variables for Real Data Examples

Table 7: Variables of the `Cars93` data

Name	Description
$PRICE$	Midrange Price (in \$1,000)
MAN	Manufacturer
	(Factor with levels "1" = Acura, "2" = Audi, ...)
$TYPE$	Type of the car
	(Factor with levels "1" = Compact, "2" = Large, "3" = Midsize,
	"4" = Small, "5" = Sporty and "6" = Van)
$MPG.C$	City MPG (miles per US gallon by EPA rating)
$MPG.H$	Highway MPG (miles per US gallon by EPA rating)
$AIRBAGS$	Standard air bags
	(Factor with levels "1" = driver and passenger, "2" = driver only,
	"3" = none)
DT	Drive train type
	(Factor with levels "1" = 4WD, "2" = front wheel, "3" = rear wheel)
$CYLIN$	Number of cylinders
$ENG.SIZE$	Engine displacement size in liters
HP	Maximum horsepower
RPM	Revolutions per minute at which maximum horsepower is achieved
$REV.MILE$	Number of revolutions of the engine needed for car to travel one
	mile in its highest gear
$MAN.TRANS$	Availability of a manual transmission
	(Factor with levels "1" = no, "2" = yes)
$FUEL.TANK$	Capacity of the fuel tank in US gallons
$PASSENGERS$	Passenger capacity (number of persons)
$LENGTH$	Length of the car in inches
$WHEELBASE$	Size of the wheelbase in inches
$WIDTH$	Width of the car in inches
$TURN$	U-turn space in feet
$REAR.SEAT$	Rear seat room in inches
$LUGGAGE$	Luggage capacity in cubic feet
$WEIGHT$	Weight of the car in pounds
$ORIGIN$	Categorization of manufacturer as domestic (U.S.) or foreign
	(Factor with levels "1" = U.S., "2" = foreign)

Table 8: Variables of the `Auto` data

Name	Description
$PRICE$	Price in US-dollars
MPG	Milage
$REP78$	Repair record 1978
$HEADROOM$	Head room in inches
$TRUNK$	Trunk space in cubic feet
$WEIGHT$	Weight of the car in pounds
$LENGTH$	Length of the car in inches
$TURN$	U-turn space in feet
$DISPLACE$	Displacement in cubic inches
$GEAR$	Gear ratio
$FOREIGN$	Categorization of manufacturer as domestic (U.S.) or foreign (Factor with levels "0" = U.S., "1" = foreign)

Table 9: Variables of the `Boston` data

Name	Description
$MEDV$	Median value of owner-occupied homes in USD 1000's
$CRIM$	Per capita crime rate by town
NOX	Nitric oxides concentration in parts per 10 million
RM	Average number of rooms per dwelling
AGE	Proportion of owner-occupied units built prior to 1940
DIS	Weighted distance to five Boston employment centres
TAX	Full-value property-tax rate per USD 10,000
$PTRATIO$	Pupil-Teacher ratio by town
B	Proportion of black population
$LSTAT$	Percentage of lower status population

References

A. Alfons, C. Croux, and S. Gelper. Sparse least trimmed squares regression for analyzing high-dimensional large data sets. *The Annals of Applied Statistics*, 7(1):226–248, 2013.

F. Alqallaf, S. Van Aelst, V.J. Yohai, and R.H. Zamar. Propagation of outliers in multivariate data. *The Annals of Statistics*, 37(1):311–331, 2009.

D.A. Belsley, E. Kuh, and R.E. Welsch. *Regression Diagnostics: Identifying Influential Data and Source of Collinearity*. John Wiley & Sons, New York, 1980.

J. Friedman, T. Hastie, H. Hofling, and R. Tibshirani. Pathwise coordinate optimization. *The Annals of Applied Statistics*, 1(2):302–332, 2007.

W. Fu. Penalized regressions: The bridge versus the lasso. *Journal of Computational and Graphical Statistics*, 7(3):397–416, 1998.

D.J. Harrison and D.L. Rubinfeld. Hedonic housing prices and the demand of clean air. *Journal of Environmental Economics and Management*, 5(1):81–102, 1978.

M. Koller and W. Stahel. Sharpening wald-type inference in robust regression for small samples. *Computational Statistics and Data Analysis*, 55(8):2504–2515, 2011.

R.J.A. Little. Regression with missing X's: A review. *Journal of the American Statistical Association*, 87(420):1227–1237, 1992.

R.A. Maronna, R.D. Martin, and V.J. Yohai. *Robust Statistics*. John Wiley & Sons, Hoboken, New Jersey, 2nd edition, 2006.

M. Salibian-Barrera and V.J. Yohai. A fast algorithm for S-regression estimates. *Journal of Computational and Graphical Statistics*, 15(2):414–427, 2006.

StataCorp. *Stata: Release 13. Statistical Software*. Stata Press, College Station, Texas, 2013.

P. Tseng. Convergence of a block coordinate descent method for nondifferentiable minimization. *Journal of Optimization Theory and Applications*, (3):475–494, 2001.

S. Van Aelst, E. Vandervieren, and G. Willems. Robust principal component analysis based on pairwise correlation estimators. In Y. Lechevallier and G. Saporta, editors, *COMPSTAT 2010: Proceedings in Computational Statistics*, pages 1677–1684, Heidelberg, 2010. Physika-Verlag.

S. Van Aelst, E. Vandervieren, and G. Willems. Stahel-donoho estimators with cellwise weights. *Journal of Statistical Computation and Simulation*, 81(1):1–27, 2011.

S. Van Aelst, E. Vandervieren, and G. Willems. A stahel-donoho estimator based on huberized outlyingness. *Computational Statistics and Data Analysis*, 56(3):531–542, 2012.

V. Verardi and C. Croux. Robust regression in Stata. *Stata Journal*, 9(3):439–453, 2009.

FACULTY OF ECONOMICS AND BUSINESS
Naamsestraat 69 bus 3500
3000 LEUVEN, BELGIË
tel. + 32 16 32 66 12
fax + 32 16 32 67 91
info@econ.kuleuven.be
www.econ.kuleuven.be